

DETAILED ACTION

1. This communication is responsive to Amendment filed 12/12/2007.

As a result of the amendment, claims 1, 46, 53, 55, 61 have been amended, claims 63-66 have been added. Claims 1-11, 22-34, 46-66 are pending in the application.

2. The objection of claims 53, 61 has been withdrawn in view of the amendment.
3. The rejection of claims 46, 55 under 35 U.S.C. §101 first paragraph has been withdrawn in view of the amendment.

EXAMINER'S AMENDMENT

4. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it **MUST** be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Applicant's representative, Mark R. Hennings, on March 11, 2008.

The application has been amended as follows:

AMENDMENTS TO THE CLAIMS:

- **Cancel claims 7, 8, 9.**
- **Claim 1 has been amended as:**

A computerized method of managing a file system for a file server, comprising:

receiving a file operation that signals a reservation operation for reserving an additional number of blocks for storing a file of the file system, the file having a file size;

computing a first number of blocks needed to accommodate the file size;

subtracting from the first number of blocks a second number of blocks already allocated for the file and a third number of delayed allocated blocks for the file to obtain a fourth number of unallocated blocks needed to accommodate the file size; and

using the fourth number of blocks to perform a reservation of unallocated blocks for the file for later allocation, wherein said using the fourth number of blocks to perform a reservation of unallocated blocks for the file for later allocation comprises:

checking that a number of available blocks in the file system is greater than the fourth number of blocks, wherein an error is returned in a case that the number of available blocks is less than the fourth number of blocks,

wherein the number of available blocks in the file system is determined by subtracting a number of allocated blocks, a number of cached unallocated blocks, and a number of reserved blocks from a total number of blocks in the file system, and adding a number of reserved cached unallocated blocks;

wherein the file system uses a write anywhere file system layout characterized in that data to be written are written to new blocks instead of being written to blocks previously allocated for said data.

- **Cancel claims 25, 30, 31.**
- **Claim 24 has been amended as:**

A file server comprising:

a memory storing a computer program,

a processor capable of executing the program, and

a storage device capable of storing files of a file system under control of the processor,
wherein the computer program comprises:

instructions to cause the processor to receive a file operation that signals a reservation operation for a file of the file system, the file having a file size;

instructions to cause the processor to compute a first number of blocks needed to accommodate the file size;

instructions to cause the processor to subtract from the first number of blocks a second number of blocks already allocated for the file and a third number of delayed allocated blocks for the file to obtain a fourth number of unallocated blocks to be reserved to accommodate the file size; and

instructions to cause the processor to use the fourth number of blocks to perform a reservation of unallocated blocks for the file for later allocation,

wherein the instructions to cause the processor to use the fourth number of blocks to perform a reservation of unallocated blocks for the file for later allocation program comprise instructions to cause the processor to check whether a number of available blocks in the file system is greater than the fourth number of blocks, and return

Art Unit: 2167

an error in a case that the number of available blocks is less than the fourth number of blocks.

wherein the processor determines the number of available blocks in the file system by subtracting a number of allocated blocks, a number of cached unallocated blocks, and a number of reserved blocks from a total number of blocks in the file system, and adding a number of reserved cached unallocated blocks;

wherein the file system uses a write anywhere file system layout characterized in that data to be written are written to new blocks instead of being written to blocks previously allocated for said data.

- **Cancel claims 47, 48, 49.**
- **Claim 46 has been amended as:**

A method comprising:

receiving at a storage server a request for a space reservation for a data set managed by the storage server; and

in response to the request,

computing a number of blocks needed to be reserved for the data set,

reserving for later allocation a number of unallocated blocks equal to the computed number of blocks, such that a subsequent write operation associated with the reservation can complete the write request without prevention of completion of the write operation due to insufficient memory; and

Art Unit: 2167

performing a write operation to write data to the data set by
determining whether a space reservation has been performed for the data
set, and

in response to determining that a space reservation has been
performed for the data set, allocating one or more blocks for said data
without determining whether enough blocks are available for completing
the write operation, and

in response to determining that a space reservation has not been
performed for the data set, determining whether enough blocks are
available for completing the write operation prior to allocating any blocks
for said data;

wherein the storage server employs a methodology in which data to be written
are written to new blocks instead of being written to blocks previously allocated for said
data.

- **Cancel claims 56, 57.**
- **Claim 55 has been amended as:**

A storage server comprising:

a processor;
a network interface through which to communicate with a remote client;
a file system; and

a storage device storing code which, when executed by the processor, causes the storage server to execute a process that includes

receiving a signal corresponding to a request for a space reservation operation for a file in the file system;

computing a first number of blocks needed to be reserved to accommodate the file;

reserving for later allocation a fourth number of unallocated blocks in the file system such that the fourth number is calculated by subtracting from the first number of blocks a second number of blocks already allocated for the file and a third number of delayed allocated blocks for the file; and

performing a write operation to write data to the file by

determining whether a block reservation has been performed for the file,

in response to determining that a block reservation has been performed for the file, allocating one or more blocks for said data in the file system without determining whether enough blocks are available in the file system for completing the write operation; and

in response to determining that a block reservation has not been performed for the file, determining whether enough blocks are available in the file system for completing the write operation prior to allocating any blocks for said data in the file system;

wherein the file system employs a methodology in which data to be written are written to new blocks instead of being written to blocks previously allocated for said data.

- **Claim 63 has been amended as:**

A computerized method of managing a file, comprising:

receiving a write request for a file;

determining a desired number of blocks in which to store the file;

determining a number of previously reserved blocks by adding allocated and nonallocated blocks previously associated with the file;

determining an additional number of blocks by subtracting the number of previously reserved blocks from the desired number of blocks in which to store the file;

reserving the additional number of blocks; and

writing the file after successfully reserving the additional number of blocks by

determining whether a space reservation has been performed for the data set, and

in response to determining that a space reservation has been performed for the data set, allocating one or more blocks for said data without determining whether enough blocks are available for completing the write operation.

in response to determining that a block reservation has not been performed for the file, determining whether enough blocks are available in the file system for completing the write operation prior to allocating any blocks for said data in the file system;
wherein the file system employs a methodology in which data to be written are written to new blocks instead of being written to blocks previously allocated for said data.

Reasons for Allowance

5. Claims 1, 3-6, 9-11, 22-24, 26-29, 32-34, 46, 50-55, 58-66, now renumbered as 1-34, are allowable over the prior arts of record.
6. The following is a statement of reasons for the indication of allowable subject matter:

The present invention is directed to system and method of managing a file system for a server to ensure that enough blocks are reserved for a file so as to ensure that the entire file can be written to a file system, without actually allocating disk blocks to the file.

Claim 1 recites, or similarly recites, in combination with the remaining elements, the steps of:

*the number of available blocks in the file system is determined by
subtracting a number of allocated blocks, a number of cached unallocated*

blocks, and a number of reserved blocks from a total number of blocks in the file system, and adding a number of reserved cached unallocated blocks;

wherein the file system uses a write anywhere file system layout characterized in that data to be written are written to new blocks instead of being written to blocks previously allocated for said data.

The closest prior art, Owada et al. (U.S. Pat. 6,185,665) shows a substantially similar file management apparatus includes a data block management device for dividing storage regions of the respective storage media into data blocks of fixed capacity and managing the use states of the divided data blocks. A data block allocation device allocates unused data blocks shown by the data block management device to a file to which a writing request from the information processor has been given (Summary). Cooper et al. (US Pat. No. 6,055,547) discloses a method and apparatus for managing the allocation and release of memory space within a number of files shared by a number of hosts in a data processing system. Guenther (US Pat. No. 5,109,336) shows data processing systems that relates to processes for memory space allocation and deallocation. However, the cited arts, singularly or in combination, fails to anticipate or render the above cited limitations in combination with other claimed elements obvious.

Claim 24 is a file server performs the method of claim 1, claim 24 is allowed under the same reasons as provided in claim 1.

Claim 46 recites, or similarly recites, in combination with the remaining elements, the steps of:

reserving for later allocation a number of unallocated blocks equal to the computed number of blocks, such that a subsequent write operation associated with the reservation can complete the write request without prevention of completion of the write operation due to insufficient memory; wherein the storage server employs a methodology in which data to be written are written to new blocks instead of being written to blocks previously allocated for said data.

The closest prior art, Sindhu et al. (U.S. Pat. 6,493,347) shows a memory organization in a switching device, and Cooper et al. (US Pat. No. 6,055,547) discloses a method and apparatus for managing the allocation and release of memory space within a number of files shared by a number of hosts in a data processing system; but both Sindhu et al. and Cooper et al., singularly or in combination, fail to anticipate or render the above cited limitations obvious.

Claim 55 recites, or similarly recites, in combination with the remaining elements, the steps of:

reserving for later allocation a fourth number of unallocated blocks in the file system such that the fourth number is calculated by subtracting from the first number of blocks a second number of blocks already allocated for the file and a third number of delayed allocated blocks for the file;

in response to determining that a block reservation has been performed for the file, allocating one or more blocks for said data in the file system without determining

whether enough blocks are available in the file system for completing the write operation.

The closest prior art, Cooper et al. (US Pat. No. 6,055,547) discloses a method and apparatus for managing the allocation and release of memory space within a number of files shared by a number of hosts in a data processing system. Owada et al. (U.S. Pat. 6,185,665) shows a substantially similar file management apparatus includes a data block management device for dividing storage regions of the respective storage media into data blocks of fixed capacity and managing the use states of the divided data blocks. A data block allocation device allocates unused data blocks shown by the data block management device to a file to which a writing request from the information processor has been given. However, both Cooper et al. and Owada et al., singularly or in combination, fail to anticipate or render the above cited limitations obvious.

Claim 63 recites, or similarly recites, in combination with the remaining elements, the steps of:

writing the file after successfully reserving the additional number of blocks by

determining whether a space reservation has been performed for the data set, and

in response to determining that a space reservation has been performed for the data set, allocating one or more blocks for said data without determining whether enough blocks are available for completing the write operation.

in response to determining that a block reservation has not been performed for the file, determining whether enough blocks are available in the file system for completing the write operation prior to allocating any blocks for said data in the file system;

wherein the file system employs a methodology in which data to be written are written to new blocks instead of being written to blocks previously allocated for said data.

The above features in conjunction with other claimed elements are distinct features which are not found in the prior art either singularly or in combination.

7. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance".

Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Miranda Le whose telephone number is (571) 272-4112. The examiner can normally be reached on Monday through Friday from 8:30 AM to 5:00 PM.

Art Unit: 2167

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John R. Cottingham, can be reached on 571-272-7079. The fax number to this Art Unit is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is 571-272-3900.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Miranda Le/

Primary Examiner, Art Unit 2167

March 13, 2008